Die Rechenmaschinen von Konrad Zuse –
Sechzig Jahre Computergeschichte

Raúl Rojas

Report B 96-

Kurzfassung

Vor sechzig Jahren wurde der mechanische Speicher der Rechenmaschine Z1 fertiggestellt. Konrad Zuses Z1 und Z3, zwischen 1936 und 1941 gebaut, bestechen noch heute durch ihre Eleganz. Beide Maschinen hatten denselben logischen Aufbau und waren die ersten vollautomatischen, programmgesteuerten Rechenmaschinen der Welt.
Das Papier beschreibt die Architektur von beiden Maschinen.

(Erscheint in *Spektrum der Wissenschaft*)

Raúl Rojas
Freie Universität Berlin
Institut für Informatik
Takustr. 9
14195 Berlin

Die Rechenmaschinen von Konrad Zuse – Sechzig Jahre Computergeschichte

Vor sechzig Jahren wurde der mechanische Speicher der Rechenmaschine Z1 fertiggestellt. Konrad Zuses Z1 und Z3, zwischen 1936 und 1941 gebaut, bestechen noch heute durch ihre Eleganz. Beide Maschinen hatten denselben logischen Aufbau und waren die ersten vollautomatischen, programmgesteuerten Rechenmaschinen der Welt.

Raúl Rojas

Die Bestimmung des genauen Geburtsorts und Geburtsdatums des Computers ist ein bis heute sehr umstrittenes Thema. In den USA wurde 1995 mancherorts über "fünfzig Jahre Computer" gesprochen – gemeint ist damit freilich die offizielle Einweihung des ENIAC (Electronical Numerical Integrator and Computer) an der Universität von Pennsylvania im Jahre 1945. Dieser elektronische Rechner war der schnellste seiner Zeit und für die Presse über Jahre hinweg der Inbegriff des ersten Computers der Welt. Die Arbeit von anderen Forschern wie John Atanasoff in Iowa oder Konrad Zuse in Deutschland wurde lange Zeit einfach ignoriert.

Was später in der Praxis realisiert wurde, war auf dem Papier bereits von dem englischen Mathematiker Alan Turing vorweggenommen worden. Der erste Entwurf seines bahnbrechenden Aufsatzes über "berechenbare Zahlen" wurde Mitte 1936 fertig. Dort schlug Turing eine operationelle Definition des intuitiven Begriffs der Berechenbarkeit vor: Nur solche Zahlen, die mit Hilfe einer Turing-Maschine berechnet werden können, gelten als überhaupt berechenbar. Eine Turing-Maschine besteht aus einem Speicher und einem kleinen "Prozessor", der nur einfachste Umwandlungen von Nullen und Einsen bewirken kann. Turing gelang es zu zeigen, daß es eine Maschine gibt, die als Universalrechner arbeiten kann. Der Speicher sollte dafür zwei Arten von Angaben enthalten: die *Daten* des zu lösenden Problems und das was wir heute das *Programm* der Berechnung nennen würden.

Wie wir in diesem Artikel zeigen möchten, war die Struktur zweier Rechenmaschinen von Konrad Zuse, der Z1 und Z3, sehr modern im heutigen Sinne. Der Fachmann erkennt an ihnen viele Konzepte und Strukturen, die in heutigen Computern fast selbstverständlich geworden sind, wie z.B. die Trennung (wie bei Turing) zwischen Prozessor und Speicher. Es kann nur verwundern, daß dies alles bereits 1936 im Ansatz von Konrad Zuse erdacht war. In jenem Jahr nämlich wurde der mechanische Speicher seines ersten Rechners fertig, so daß wir, wegen der Arbeiten von Turing und Zuse, 1996 eigentlich über das sechzigjährige Jubiläum des *Konzeptes* des Computers (allerdings nicht des Computers selbst) sprechen können.

Konrad Zuse war kein Mathematiker und hatte sich mit dem Berechenbarkeits-Begriff in seiner mathematischen Fassung nie auseinandergesetzt. Aus der Praxis kommend, interessierte ihn eher die Möglichkeit, lange Ketten von langweiligen Berechnungen voll zu automatisieren. Seine breite Ausbildung als Architekt und Bauingenieur, sowie seine ganz besondere Gabe, komplexe Automaten aus einfachsten Teilen herstellen zu können, führten ihn sehr früh auf die Idee eines programmierbaren Automaten. Sein erster Versuch war die Rechenmaschine Z1, ein praktisch vollständig mechanisches Gerät, das dafür gedacht war, die vier arithmetischen Rechenoperationen (Addition, Subtraktion, Multiplikation und Division) in beliebiger Reihenfolge und mit gespeicherten Zahlen auszuführen. Die Z1 wurde 1938 fertiggestellt, aber die mechanischen Bauteile (bewegliche Bleche) waren nicht zuverlässig genug, so daß Zuse auf Relaistechnik umstieg. Nach einigen Experimenten mit einer hybriden Maschine (Z2) baute Zuse die Z3, die 1941 fertig und funktionsfähig war. Es ist anzumerken, daß die Z3, vom logischen Standpunkt, äquivalent zur Z1 war. Die dafür gewählten numerischen Algorithmen und die Kodierung der Zahlen waren dieselben – die Z3 war die zuverlässige Z1. Es hat 5 Jahre gedauert, bis Zuse seine Vision von 1936 realisieren konnte, eine erstaunlich kurze Zeit, wenn man bedenkt, daß er völlig auf sich selbst gestellt war, anders als die großen Gruppen, die in den USA an der ENIAC oder anderen Maschinen mit praktisch unbegrenzter Finanzierung gearbeitet haben.

In diesem Artikel wollen wir die logische Struktur der Z1 und Z3 anhand der Patentanmeldung, die Zuse 1941 für die Z3 einreichte, beleuchten. Wie wir sehen werden, war die Z3 eine erstaunlich moderne Maschine, ein Höhepunkt der damaligen Technik. Es ist kaum eine bessere Möglichkeit vorstellbar, an das Lebenswerk des verstorbenen Konrad Zuse zu erinnern, als ins Innere dieser Maschine zu steigen und ihre Geheimnisse offenzulegen. Nur so kann die große Leistung des deutschen Erfinders gebührend gewürdigt werden.

Die Grundstruktur der Z3

Die Z3 wurde mit Relais gefertigt. Die Maschine bestand aus zwei Hauptteilen, dem Rechenwerk und dem Speicher. Für den ersteren wurden 600 Relais benötigt, für den letzteren 1400. Der Speicher war für 64 Zahlen ausgelegt und hatte eine regelmäßige Struktur, war also einfach zu realisieren. Das Rechenwerk dagegen war extrem kompakt und in vielerlei Hinsicht optimiert worden. Es konnte die vier arithmetischen Operationen ausführen sowie die Quadratwurzel einer Zahl berechnen. Die Zahlen wurden in Basis 2 kodiert, und dafür wurden im Speicher 22 Bit verwendet. Die Steuerung der Maschine geschah über einen 8-Kanal-Lochstreifen, in dem eine Folge von arithmetischen Operationen angegeben werden konnte. Nach dem Start konnte die Maschine die Berechnungen vollautomatisch bis zum Ende ausführen. Sie war allerdings, nach heutigen Maßstäben, nicht sehr schnell, da eine Multiplikation um die 3 Sekunden benötigte.

Es ist interessant zu konstatieren, daß die Wahl der Basis 2 für die Darstellung der Zahlen damals keineswegs die unangefochtene technische Lösung war. Aiken hat für seinen Rechner Mark I, der von 1939 bis 1944 in Harvard gebaut wurde, eine mit der Basis 10 arbeitende mechanische Lösung verwendet. Die ENIAC, obwohl vollelektronisch, verwendete eine Kette von 10 Vakuumröhren, um eine Ziffer von 0 bis 9 zu kodieren (nur eine der Röhren wurde für jede Ziffer eingeschaltet). Da Zuse für die Z1 mechanische Elemente (in zwei Richtungen bewegliche Bleche) verwendete, war die Binärdarstellung einfacher. Zudem, dachte Zuse, sind in der Maschine die Zahlen "unter sich", und der Mensch braucht den einzelnen Berechnungsschritten nicht folgen zu können: es reicht, wenn das Resultat korrekt ist.

Zuse hat aber mehr getan als nur die Basis für die Darstellung von ganzen Zahlen von 10 auf 2 zu verändern: er hat die heutige Gleitkomma-Notation erfunden. Binäre Zahlen können, ebenso wie Dezimalzahlen, mit Ziffern vor und nach dem Komma geschrieben werden. Die Stellen vor dem Komma stellen die Potenzen 2^0, 2^1, 2^2 usw. dar, die Stellen nach dem Komma stellen die Potenzen 2^{-1}, 2^{-2} usw. dar. Jede so geformte Mantisse wird außerdem mit einer Potenz von 2 multipliziert, ähnlich wie wir Dezimalzahlen in der wissenschaftlichen Notation schreiben oder in Taschenrechner eingeben (siehe Kasten: "Die Gleitkomma-Notation").

Während die Zahlen im Speicher mit 14 Binärstellen nach dem Komma dargestellt wurden, arbeitete das Rechenwerk mit einer zusätzlichen Genauigkeit, nämlich zwei Extrabits am Ende der Stellen nach dem Komma. Diese zusätzliche Genauigkeit erlaubt exakte Rundung der Resultate der arithmetischen Operationen; es ist dieselbe Technik, die in heutigen Computern verwendet wird.

Zahlen wurden manuell über eine spezielle Tastatur mit maximal 4 Dezimalziffern in die Maschine eingegeben. Die zugehörige Potenz von 10 wurde auch eingegeben. Die Zahl 4,356 z.B. wurde über die Tastatur als die Zahl 4356 eingestellt. Der Exponent von 10 wurde als –3 über eine Kommaeingabevorrichtung eingegeben. Das Rechenwerk verwandelte die Dezimaldarstellung automatisch in die binäre, so daß der Rechner intern mit diesen Zahlen weiter operierte. Dem Programmierer standen 9 Befehle zur Verfügung: davon hatten zwei mit der Ein-Ausgabe, zwei mit dem Laden und Lesen des Speichers und fünf mit arithmetischen Operationen zu tun. Dem Programmierer standen zwei Register (rechenwerkinterne Speicherzellen) zur Verfügung. Befehle in beliebiger Reihenfolge konnten mit den dort gespeicherten Zahlen und mit den Registern operieren.

Programmierung der Z3

Das Programmiermodell der Z3 (d.h. der für den Programmierer sichtbare Teil der Architektur) besteht aus zwei Gleitkomma-Registern: Register 1 und Register 2. Der erste Lade-Befehl (von der manuellen Eingabe oder vom Speicher) lädt Register 1. Jeder weitere Lade-Befehl überschreibt Register 2. Resultate der arithmetischen Operationen werden in Register 1 geschrieben, und Register 2 wird dabei gelöscht. Solange Register 1 nicht gelöscht wird, beziehen sich alle Lade-Befehle auf Register 2.

Register 1
Register 2

Exponent: 7 Bit
Mantisse: 15 Bit

Der Befehlssatz der Z3 umfaßt folgende 9 Befehle:

Eingabe-Ausgabe:

Lu	Von der Eingabe lesen
Ld	Register 1 in die Ausgabe leiten (Anzeigen)

Speicher-Befehle:

Pr z	Vom Speicher lesen (Adresse z)
Ps z	Register 1 speichern (Adresse z)

Arithmetische Befehle:

Lm	Multiplikation
Li	Division
Lw	Quadratwurzel
Ls1	Addition
Ls2	Subtraktion

Mit Ausnahme der Operation Quadratwurzel haben alle arithmetischen Operationen zwei Argumente. Quadratwurzel bezieht sich auf das Argument in Register 1. Nach jeder Speicheroperation (z.B. Ps 10) müssen beide Register neu geladen werden, d.h. eine Speicheroperation löscht Register A und B. In der Z3 hat Register A die Adresse 0, d.h. der Befehl "Ps 0" hat keinen Effekt.

Die Blockarchitektur der Z3

Die wichtigsten funktionellen Teile der Z3 sind: der Speicher, das Rechenwerk, die Kontrolleinheit für die einzelnen Befehle (Leitwerk), die Zahleneingabe und die Zahlenausgabe. Bild 1 zeigt ein Diagramm des Lochstreifenlesers und der Dekodiereinheiten (Schaltung Pa und Pb) sowie des Leitwerks (die Einheit für die Steuerung der Maschine) und der Ein-Ausgabe-Einheiten. Jede Zeile des Lochstreifens wird in einen Befehl verwandelt, dessen Einzelschritte vom Leitwerk gesteuert werden. Lade- und Schreib-Befehle vom und zum Speicher werden in der Schaltung Pb dekodiert, der die Adresse der entsprechenden Speicherzelle bereitstellt.

Die Schaltung V des Leitwerks steuert die Berechnung der Vorzeichen des Resultats von arithmetischen Operationen; die Schaltung U die Verwandlung der eingegebenen Dezimal- in eine Binärzahl und die Schaltung D die Rückübersetzung von Binär in Dezimal sowie die Anzeige des Resultats im Lampenfeld R (Kommastellung in Schaltung Q).

Heutige Computer teilen die Abarbeitung der Befehle in einzelne Stufen, die jeweils einen Maschinenzyklus benötigen. Die Z3 arbeitet mit einem Taktgeber, der jeden Zyklus oder Takt (Zuse nannte ihn ein "Spiel") in fünf Schritte (I, II, III, IV und V) unterteilt.

Das Grundmuster für die Bearbeitung der Befehle ist folgendes: im ersten Zyklus wird der Befehl dekodiert (Schritt I). Operanden für die arithmetischen Befehle werden dann in Schritt IV und V vorbereitet und an die entsprechenden Schaltungen der Z3 geführt. Im zweiten Zyklus kann in den Schritten I, II und III eine Addition/Subtraktion der Exponenten und Mantissen berechnet werden. In den Schritten IV und V werden die Resultate in die richtigen Register zurückgeschrieben (Bild 2).

Bei Befehlen, die mehrere Takte benötigen, z.B. Multiplikation und Division, wird in Takt 2 der Lochstreifenmechanismus und die Dekodierung weiterer Befehle gestoppt. Ist der Befehl abgearbeitet, wird die Dekodierung weiterer Befehle freigegeben.

Der Befehl Pr (aus dem Speicher laden) kann in einem einzigen Zyklus ausgeführt werden, der Befehl Ps (speichern) kann mit dem letzten Takt eines vorherigen arithmetischen Befehls überlappt werden und somit sogar in "Nullzeit" ausgeführt werden.

Die Aufgabe des Leitwerks ist es, die richtige Steuerung des Rechenwerks zu ermöglichen. Dafür wird für mehrstufige Befehle ein Steuerrad gestartet, das die richtigen Relais im Rechenwerk im richtigen Moment schließt oder öffnet. Diese Steuerräder entsprechen dem Mikroprogramm heutiger Prozessoren, das jeden Befehl auf eine Folge von elementaren Mikrobefehlen zurückführt. An der Peripherie des Steuerrades sind Kabel angebracht, ein (am Strom angeschlossener) Schalter kreist auf dem Steuerrad und bewegt sich eine Stelle pro Takt. Auf diese Weise können die Mikrobefehle an das Rechenwerk weitergegeben werden.

Bild 3 zeigt eine vereinfachte Darstellung des Rechenwerks. Es besteht aus zwei Teilen: Teil A führt die Exponenten-Operationen aus, Teil B arbeitet mit den Mantissen. Das Gleitkomma-Register 1 des Programmiermodells entspricht eigentlich zwei Registern im Rechenwerk: Register Af für den Exponenten und Register Bf für die Mantisse. Gleitkomma-Register 2 des Programmiermodells entspricht Register Ab für den Exponenten und Register Bb für die Mantisse. Das Register Af kann über entsprechende Leitungen auf Aa oder Ab kopiert werden. Die in Bf gespeicherte Mantisse kann auf Ba oder Bb weitergegegeben werden, wenn gewünscht sogar verschoben (über die gezeichneten Shifter).

Die Kästchen mit den Aufschriften Ea, Eb, Ec, Ed, Ef und Fa, Fb, Fc, Fd, Ff sind Durchlaßschaltungen. Sind die entsprechenden Relais geschlossen, fließt keine Information durch die Schalter, sind sie offen, können die entsprechenden nachgeschalteten Register geladen werden. Das A und O der richtigen Steuerung des Rechenwerks ist, diese Durchlaßschalter aufeinander abzustimmen und im richtigen Moment zu öffnen oder zu schließen. So fließt die Information über die gewünschten Bahnen. Die zwei gezeigten Addierer-Subtrahierer (A und B) haben die Aufgabe, Exponenten bzw. Mantissen in einem Zyklus zu addieren oder zu subtrahieren, je nach Operation. Die Resultate werden auf die Ae- und Be-Register weitergegeben; von dort können sie rezirkuliert (über die Aa, Ba-Schaltungen) oder auf Af, Bf zurückgeführt werden.

Der Übersichtlichkeit wegen haben wir viele der Kontrolleitungen von der Darstellung ausgespart.

Die numerischen Algorithmen

Die Z3 arbeitet mit Taktgebern, die verschiedene numerische Operationen über mehrere Zyklen kontrollieren können. Ein Beispiel dafür ist die Multiplikation der zwei Gleitkomma-Zahlen $a \times 2^b$ und $c \times 2^d$. Der dafür verwendete Algorithmus ähnelt der üblichen Schulmultiplikation.

Im Teil A der Z3 (Exponententeil) können die Exponenten in einem Zyklus addiert werden. Teil B der Z3 (Mantisseteil) braucht so viele Zyklen wie die Anzahl der Bits in dem Multiplikator.

Ein Beispiel mit 4 Bits soll die Methode illustrieren. Nehmen wir an, die Mantisse des

Multiplikators sei 1,101 und die des Multiplikands 1,001. Die vier Bits des Multiplikators werden eins nach dem anderen (von rechts nach links) mit dem

Multiplikanden multipliziert und eine Stelle nach rechts verschoben. Das Ergebnis der Multiplikation ist dann

```
   1001
  0000
 1001
1001
-------
1110101
```

Da in diesem Beispiel die Maschine nur 4 Bits verarbeitet werden, besteht das Resultat aus den ersten 4 Bits mit dem Komma an der üblichen Stelle, also 1,110. Die Berechnung kann in eine Reihe von 4 Additionen in 4 Schritten verwandelt werden. In Schritt 0 wird das vorläufige Resultat auf 0000 gesetzt. In jedem weiteren Schritt wird der Multiplikand addiert, falls das Bit des Multiplikators 1 ist, sonst nicht. Das vorläufige Resultat wird um eine Stelle nach rechts verschoben. Das Endresultat entsteht schließlich in Schritt 4 und wird nicht verschoben (siehe Bild 4).

Die Division wird ähnlich berechnet, nur daß jetzt in jedem Schritt eine Subtraktion von Mantissen und eine Verschiebung vorgenommen werden. Der Quotient wird iterativ, Bit für Bit, aufgebaut. Der Quadratwurzel q von einer Zahl x wird ebenfalls iterativ aufgebaut, und zwar so, daß $x/q = q$. Dies bedeutet, daß der Quadratwurzeloperation eine ähnliche Strategie wie die einfache Division zugrunde liegt.

Da die Z3 intern mit 15+2, also 17 Bits rechnet, benötigt eine Multiplikation 17 Zyklen (die zwei Extra-Bits sollen eine erhöhte Genauigkeit bei Berechnungen garantieren). Dies entspricht etwa 3 Sekunden. In jedem Zyklus wird eine Addition durchgeführt; das Resultat wird wieder als Argument für die nächste Addition vorbereitet. Der Multiplikand bleibt als zweites Argument im Register B des Mantisseteils. Ähnlich wird die Division und die Quadratwurzelberechnung durchgeführt. Die grundlegende Operation der Z3 ist deswegen die Addition von Exponenten und Mantissen. Eine solche Addition kann in einem Zyklus, wie bei heutigen Mikroprozessoren, berechnet werden.

Vergleich der Z3 mit anderen Rechnern

Es ist interessant, die technischen Daten der Z3 mit denen anderer Rechner zu vergleichen, die in den USA etwa zur gleichen Zeit gebaut wurden. Dazu zählen die ABC-Maschine von John Atanasoff, gebaut von 1938 bis 1942, Mark I der Universität Harvard, gebaut von 1939 bis 1944 und ENIAC, gebaut an der Universität von Pennsylvania von 1943 bis 1945.

Tabelle 1 zeigt die wichtigsten architektonischen Unterschiede zwischen diesen drei Maschinen und der Z3. Ein wichtiges Merkmal heutiger Computer ist die Trennung von Prozessor und Speicher. Mark I und ENIAC haben diese Trennung nicht vollzogen. Die Speicherzellen dienten zugleich als arithmetische Elemente, d.h. die Maschinen waren viel zu komplex in Bezug auf ihre numerische Leistung. Nur die Z3 und ABC waren Binärmaschinen. Mark I und ENIAC benutzten eine interne Dezimalkodierung, obwohl die Arithmetik teilweise binär berechnet wurde. Keine der Maschinen bot dem Programmierer bedingte Sprünge im Befehlssatz. Dies war sicherlich einer der gravierendsten Mängel der Z3. Ohne bedingten Sprung ist es nicht möglich, kompliziertere Algorithmen zu programmieren, die Verzweigungen im Kontrollfluß verwenden können. Im Grunde genommen kann eine Rechenmaschine, die keine bedingten Sprünge verarbeiten kann, keineswegs als Universalrechner gelten. Die ENIAC konnte Schleifen abbrechen und den Kontrollfluß weiterleiten, aber dafür stand nur eine Kontrolleinheit zur Verfügung.

Tabelle 1 zeigt auch, daß nur die Z3 die Gleitkommakodierung verwendet hat. Die Z3 und die Mark I waren elektromechanische Rechner, während die ABC und die ENIAC mit Vakuumröhren arbeiteten. Die ABC war für die Arbeit mit Vektoren ausgelegt, deswegen können wir sie als einen Vorläufer der modernen Vektorrechner bezeichnen. Die ENIAC dagegen hatte eine Datenfluß-Architektur, d.h. Hardware-Bausteine wurden fest verdrahtet, und ihre Kombination definierte die auszuführende Berechnung. Weder für die ENIAC noch für die ABC wurden Programme geschrieben. Die ABC konnte sogar nur ein einziges festes Programms (Gauß-Elimination) ausführen. Die Z3 und die Mark I wurden durch Programme auf Lochstreifen gesteuert.

Aus der Perspektive von Tabelle 1 kann man gut erkennen, daß die Z3 ein relativ moderner Rechner war. Sein größtes Manko war, daß bedingte Sprünge im Befehlssatz nicht berücksichtigt wurden. Zuse war dieses Problem bewußt, er führte aber Sprünge erst bei seinem Rechner Z4 ein, der zum Ende des Krieges fertiggestellt wurde.

Erstaunlich ist jedoch, daß die beiden Maschinen Mark I und ENIAC von mittelgroßen Gruppen von Ingenieuren gebaut wurden. Zuse und Atanasoff dagegen haben ihre Maschinen fast im Alleingang geschaffen. Dies hat beide dazu geführt, die Architektur zu optimieren und nicht mehr Hardware als wirklich notwendig zu verwenden. Die architektonischen Konzepte sind jedoch bei der Z3 viel weiter ausgereift als bei der ABC, der eigentlich ein Rechner für eine spezielle Aufgabe war. Bild 5 zeigt die Z3 in ihrer ganzen Komplexität. Die wesentlichen Strukturen und ihr Zusammenspiel sind dort gezeichnet.

Die Erfindung des Computers

Der Erfolg hat bekanntlich viele Väter, und im Fall des Computers kann man von einem langen Entwicklungsprozeß bis zur Entstehung des heutigen Universalrechners reden. Von der genialen Arbeit von Turing bis zu dem späteren Universalrechner ist der Weg mit vielen Bruchstücken von Ideen und Erfindungen gepflastert. Dabei darf auch der Beitrag eines Charles Babbage nicht vergessen werden, der mit seiner Analytischen Maschine bereits im neunzehnten Jahrhundert als Erster programmierbare Rechner zu entwerfen wagte. In diesem Jahrhundert hat John von Neumann wesentlich dazu beigetragen, die konzeptionellen Grundlagen der Rechnerarchitektur auf sichere Füße zu stellen. Obwohl von Neumann selbst keinen Rechner baute, stand er an vorderster Stelle, wenn es darum ging, Berechenbarkeit theoretisch zu erfassen oder in die Praxis einzubringen.

An dieser Stelle sei mir eine Anekdote erlaubt. Als Abschluß für einen Vortrag über die Erfindung des Computers habe ich einmal drei Namen auf die letzte Folie geschrieben, nämlich Babbage, Zuse, von Neumann. Die Folie habe ich Konrad Zuse 1993 gezeigt, da ich naiverweise dachte, er würde sich freuen, seinen Name in so illustrer Gesellschaft zu sehen. Zuse hat aber nur gefragt: "Was hat von Neumann da zu suchen?" An Selbstbewußtsein mangelte es dem genialen Erfinder sicherlich nicht, und seinen Platz in der Computergeschichte hat er sich mit seinem Lebenswerk erkämpft. Erstes Resultat davon war das mechanische Speicherwerk von 1936, das die Ära des Computers einläutete.

Literaturhinweise

Konrad Zuse, der Weg zu seinem Computer Z3. Von K.-H. Czauderna. Oldenbourg-Verlag, 1979.

On Basic Concepts of Early Computers in Relation to Contemporary Computer Architectures. Von R. Rojas in: 13th World Computer Congress 94, Hamburg 1994, Seiten 324 bis 331.

Who invented the computer? The debate from the viewpoint of computer architecture. Von R. Rojas in: W. Gautschi (Hrsg.), Fifty Years Mathematics of Computation, Proceedings of Symposia in Applied Mathematics, AMS, 1993, Seiten 36 bis 366.

Patentanmeldung Z-391. Von K. Zuse. 1941.

Der Computer – Mein Lebenswerk. Von K. Zuse. Springer-Verlag, 1984.

Die Gleitkomma-Notation

1 Bit	7 Bit	14 Bit
+/–	Exponent	Mantisse

Heutige Computer arbeiten mit Binärzahlen. Ähnlich wie bei Dezimalzahlen kann man Stellen vor und nach dem Komma verwenden. Die Dezimalzahl 18,73 ist nur die bequeme Schreibweise für den Ausdruck

$$18{,}73 = 1 \times 10^{1} + 8 \times 10^{0} + 7 \times 10^{-1} + 3 \times 10^{-2}.$$

Analog ist die Binärzahl 10,11 eine bequeme Schreibweise für

$$10{,}11 = 1 \times 2^{1} + 0 \times 2^{0} + 1 \times 2^{-1} + 1 \times 2^{-2}.$$

Konrad Zuse war der erste, der Gleitkomma-Binärzahlen in programmierbaren Rechnern verwendete. Er nannte seine Notation die "halblogarithmische Schreibweise". Die Dezimalzahl 8731 z.B. kann als das Produkt 8,731 x 10^3 geschrieben werden (wir ziehen es vor, nur eine Stelle vor dem Komma zu verwenden). Die Zahl 3 nennt man den Exponenten und 8,731 die Mantisse. Dasselbe kann mit Binärzahlen gemacht werden. Jede Zahl wird dann in der Form

$$a \times 2^{b}$$

geschrieben, wobei b eine ganze Zahl ist und a eine Binärzahl größer oder gleich Null und kleiner als 2. Ein Beispiel wäre die Binärzahl a = 1,11 mit Binärexponent b = 2. Die Binärzahl 1,11 ist die Dezimalzahl 1,75, weil $1 \times 2^{0} + 1 \times 2^{-1} + 1 \times 2^{-2} = 1{,}75$. Deswegen stellt $a \times 2^{b}$ in diesem Fall die Zahl

$$1{,}75 \text{ x } 4 = 7$$

dar. Zuse verwendete 7 Bit für den Exponenten und 15 Bit für die Mantisse. Das Vorzeichen der Zahl wurde in einem zusätzlichen Bit gespeichert. Da die Exponenten in der sogenannten Zweierkomplementdarstellung kodiert wurden, konnte man sowohl positive als auch negative Exponenten verwenden.

Zuse normalisierte jedoch alle Binärzahlen derart, daß nur ein Bit vor dem Komma verwendet wurde. Durch Anpassung des Exponenten kann dieses Bit immer gleich 1 gesetzt werden (mit Ausnahme der Zahl Null). Da dieses erste Bit immer gleich 1 ist, braucht es nicht gespeichert zu werden. Deswegen können im Speicher 14 Bit für die Mantisse verwendet werden. Die Mantisse wird auf 15 Bit bei jedem Speicherlesezugriff erweitert.

Da die Zahl Null bei dieser Konvention Probleme verursacht, behandelte Zuse jede Zahl der Form $a \times 2^{-64}$ als Null. Der Exponent –64 wurde durch Spezialschaltungen aufgefangen, und die korrekten Resultate wurden im Rechenwerk produziert. Darüber hinaus verwendete Zuse die Konvention, daß Zahlen der Form $a \times 2^{63}$ unendlich groß sind. Spezialschaltungen sorgten wiederum für die korrekte Arithmetik.

Wie weit seiner Zeit voraus die von Zuse entwickelte Kodierung war, erkennt man ohne Mühe, wenn man die heutigen IEEE-Gleitkomma-Standards betrachtet. Sie sind im wesentlichen äquivalent zu Zuses halblogarithmischer Schreibweise. Der einzige Unterschied besteht in der Anzahl der verwendeten Bits und in der Kodierung der numerischen Sonderfälle (Null, unendlich, usw.).

Beispiel eines Programms für die Z3

Nehmen wir an, daß in den Speicheradressen 1, 2, 3 und 4 die Konstanten a_1, a_2, a_3 und a_4 gespeichert sind. Weiterhin ist die Zahl x in Adresse 5 gespeichert. Die Berechnung

$$x\,(a_2 + x\,(a_3 + x\,a_4))) + a_1$$

kann mit dem folgenden Programm in der Z3 durchgeführt werden (Kommentare in geschweiften Klammern):

```
Pr  4       {Lade a4 in Register 1}
Pr  5       {Lade x in Register 2}
Lm          {Multiplikation, Resultat in Register 1}
Pr  3       {Lade a3 in Register 2}
Ls1         {Addiere Register A und B, Resultat in 1}
Pr  5       {Lade x in Register 2}
Lm          {Multiplikation, Resultat in Register 1}
Pr  2       {Lade a2 in Register 2}
Ls1         {Addiere Register A und B, Resultat in 1}
Pr  5       {Lade x in Register 2}
Lm          {Multiplikation, Resultat in Register 1}
Pr  1       {Lade a1 in Register 2}
Ls1         {Addiere a1 und x(a2 + x(a3 + xa4)))}
Ps 10       {Speichere Resultat in Adresse 10}
```

Eingabe-Ausgabe-Einheiten und Leitwerk der Z3

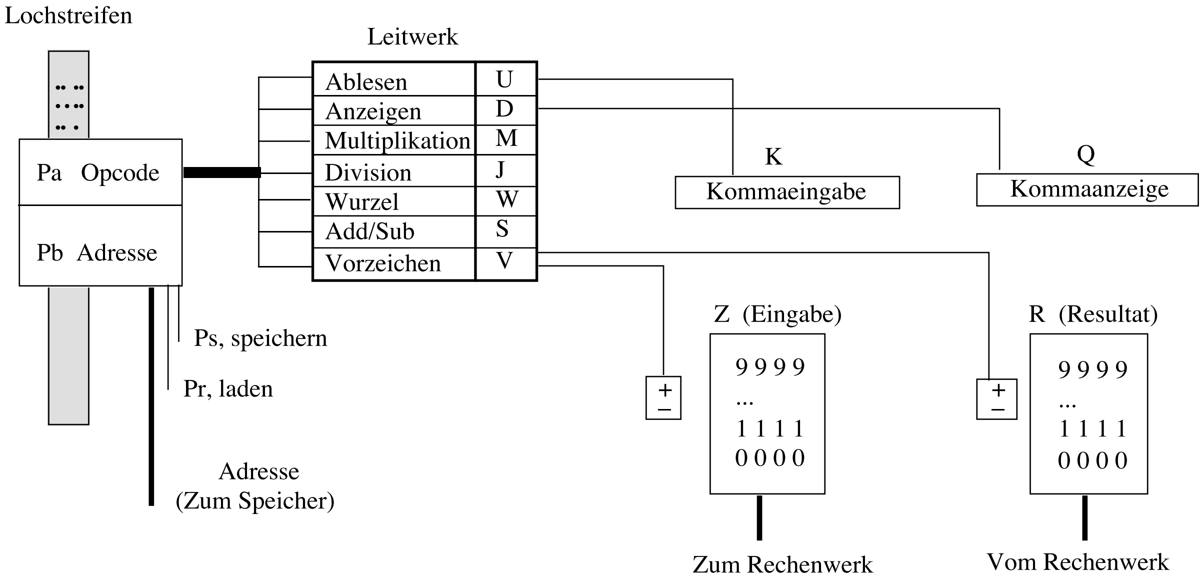

Bild 1: Das Programm wird Befehl für Befehl in einen Lochstreifen gestanzt. Die Schaltung Pa dekodiert den Befehlstyp und die Schaltung Pb die Adresse für Laden- und Speichernbefehle. Das Leitwerk steuert die Ausführung der Befehle und sorgt für die korrekte Koordinierung der Z3-Module. In der Eingabetastatur Z kann eine Dezimalzahl mit 4 Ziffern eingestellt werden. Die Potenz von 10 wird in der Kommaeingabe (Schaltung K) eingestellt. Die Ausgabe des Resultats als Dezimalzahl erfolgt über Lampen (Schaltung R); der Dezimalexponent wird in Q angezeigt.

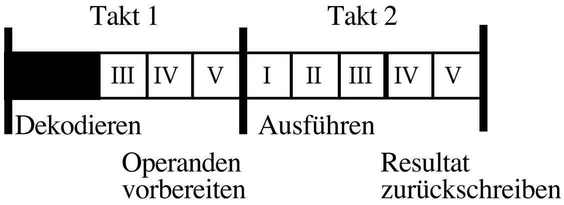

Bild 2: Der Maschinenzyklus der Z3. Die Befehle werden im ersten Takt dekodiert (Spiele I bis III). In den Spielen IV und V werden die Operanden für den Befehl vorbereitet. In jedem darauffolgenden Takt wird eine Elementaroperation ausgeführt (mit den Exponenten und/oder Mantissen). Die Resultate werden in den Spielen IV und V in die erforderlichen Module zurückgeschrieben.

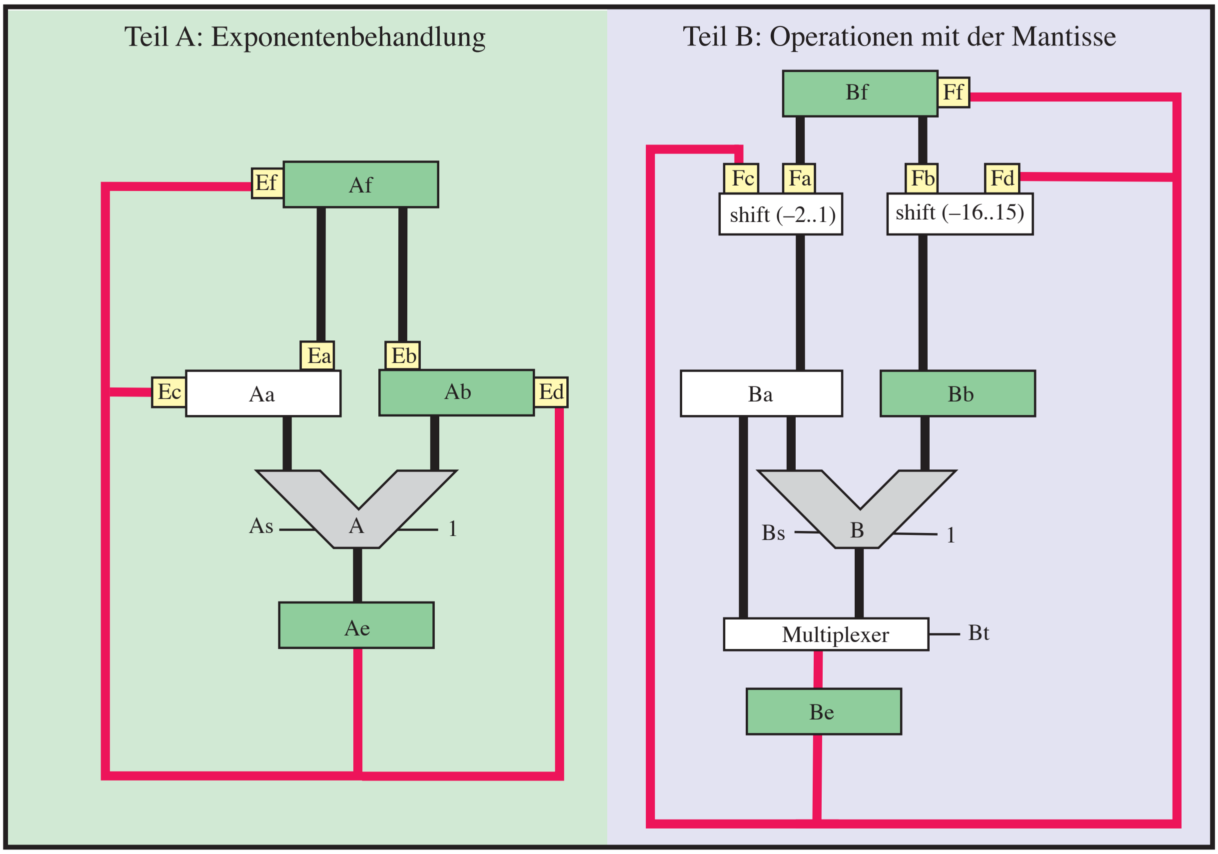

Bild 3: Vereinfachte Darstellung des Rechenwerks. Der linke Teil verarbeitet die Exponenten der Binärzahlen, der rechte die Mantissen. Die grünen Module kennzeichnen Register, die aus selbsthaltenden Relais aufgebaut sind. Sie behalten ihren Inhalt, bis sie durch eine entsprechende Leitung gelöscht werden. Die gelben Kästchen sind Durchlaßschalter, die das Laden der Register ermöglichen. Die zwei grauen Schaltungen sind Addierer-Subtrahierer. Falls As bzw. Bs gesetzt sind, wird subtrahiert. Das Resultat geht in Register Ae bzw. Be und kann von dort in die Argumentregister zurückgeleitet werden. Die zwei Shifter im Mantisseteil erlauben, die Mantisse um mehrere Stellen nach rechts oder nach links zu verschieben. Diese Operation entspricht einer Multiplikation mit negativen oder positiven Potenz von 2. Im Mantisseteil kann das Resultat der Addition/Subtraktion oder eines der Argumente durch den vor Be geschalteten Multiplexer ausgewählt werden. Diese Auswahl ist bei der Multiplikation und der Division notwendig. Beide Teilschaltungen A und B sind geschlossene Kreisläufe. Ein Durchlauf benötigt einen Takt (ein Spiel).

Produkt zweier Binärzahlen (Mantisse)

Multiplikand = 1,001
Multiplikator = 1,101

Schritt	Vorläufiges Resultat	Bit des Multiplikators
0	0000	
1	0100	1
2	0010	0
3	0101	1
4	1110	1

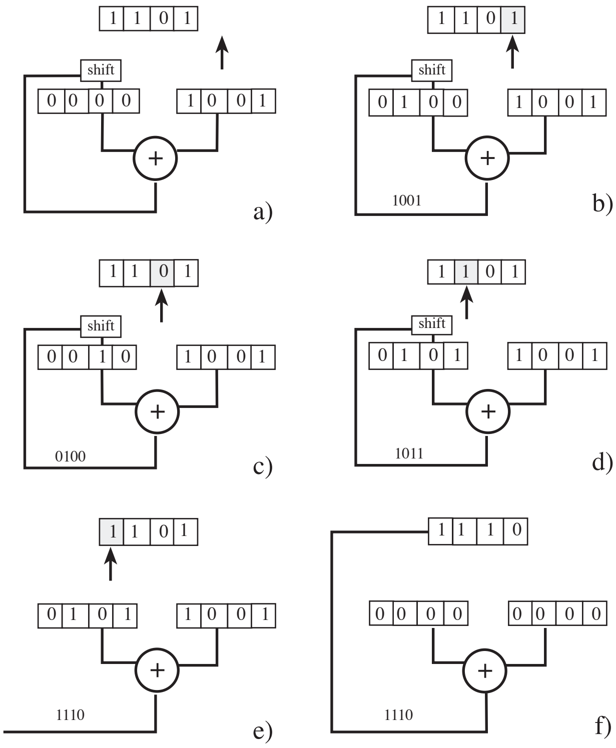

Bild 4: Produkt der Binärzahlen 1,101 und 1,001 (um die Diskussion zu vereinfachen, verwenden wir nur Register mit vier Bits).
Die Abbildung zeigt das Zusammenspiel der Z3-Register Bf, Ba und Bb bei der Berechnung der obigen Multiplikation. Diagramm a) zeigt die Anfangskonfiguration, mit dem Multiplikator in Bf und dem Multiplikanden in Bb. Die Diagramme b) bis e) zeigen die Resultate der Additions- und Shift-Operation nach jedem Takt. Der Pfeil zeigt, welches Bit des Multiplikators abgearbeitet wird. In f) wird gezeigt, wie das Resultat im letzten Takt auf Bf zurückgeführt wird (diesmal ohne shift).

Rechner	Prozessor und Speicher getrennt?	Codierung	Fließkomma?	Sprünge?	Programme	Architektur	Technologie
Z3	ja	Binär	ja	nein	Software	sequentiell	elektro-mechanisch
ABC	ja	Binär	nein	nein	nicht programmierbar	vektoriell	elektronisch
Mark I	nein	Dezimal	nein	nein	Software	sequentiell	elektro-mechanisch
ENIAC	nein	Dezimal	nein	zum Teil	Hardware	Datenfluß	elektronisch

Tabelle 1: Die Z3 im Vergleich mit anderen Rechnern. Die Z3 verwendete Binär- und Gleitkommakodierung, der Prozessor war vom Speicher getrennt, und Programme konnten geschrieben werden. Die ABC war eine elektronische Maschine mit einem festeingebauten Programm. ENIAC wurde durch die Hinter–einanderschaltung von Hardware-Modulen "programmiert". ENIAC war in diesem Sinne ein Datenflußrechner ohne Software. Schleifen konnten bei diesem Rechner mit einem bedingten Sprung abgebrochen werden.

Bild 5: Diagramm der Gesamtstruktur der Z3. Die Strukturen mit rosa Hintergrund gehören zum Leitwerk und zu den Ein-Ausgabeeinheiten. Der Speicher wurde mit einem blauen Hintergrund gezeichnet. Die Adreßleitungen werden vom Leitwerk gesetzt. Der Exponent wird in die linke Teilschaltung geladen, die Mantisse in die rechte.

Die Einzelschaltungen haben Namen, die die Relaisgruppen kennzeichnen. Die gelben Rechtecke sind Durchlaßschalter für das Laden von Registern, eventuell mit einem zusätzlichen Shift.

Einige Durchlaßschalter werden durch Ah_1, Ah_2 und Bh gesteuert. So wird das Register angesprochen, das vom Programm ausgewählt wird. Die Konstanten 13 und 4 können direkt in Register Aa bzw. Ab geladen werden (notwendig bei gewissen Algorithmen). Bevor der Exponent aus Ae in den Berechnungskreislauf weitergegeben wird, wird getestet, ob ein Sonderfall (unendliche Zahl, Null) vorliegt. Dies produziert gewisse Anpassungen der Maschine.

Im Mantisseteil des Rechenwerks (violetter Hintergrund) wird überprüft, ob die Mantisse "gerichtet" werden soll. Dies bedeutet, daß das erste Bit der Mantisse eine Eins sein muß, sonst wird die Mantisse so viele Stelle wie notwendig nach links verschoben (außer bei Null). Die Verschiebung wird durch die Leitungen Fh bis Fm angegeben. Die Verschiebung der Mantisse verändert den Exponenten der Zahl (über die Schalter Bn).

.

Raúl Rojas ist Hochschullehrer für Praktische Informatik und Künstliche Intelligenz an der Martin-Luther-Universität Halle. Er hat an der Freien Universität Berlin promoviert und habilitiert. Während seiner Zeit beim Forschungsinstitut für Innovative Rechensysteme und Technologie der GMD in Berlin beschäftigte er sich eingehend mit der Architektur von symbolischen Maschinen. Sein Interesse für die Entstehungsgeschichte des Computers hat zu verschiedenen Veröffentlichungen auf diesem Gebiet und zum Projekt, die Z3 in Software zu simulieren, geführt.